

МІНІСТЕРСТВО ОСВІТИ І НАУКИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій  
Кафедра систем штучного інтелекту



Лабораторна робота №2  
з курсу “Дискретна математика ”

Виконав:  
ст. гр. КН-110  
Петровський Олександр

Викладач:  
Мельникова Н.І.

Львів – 2018

## **Тема:**

”Моделювання основних операцій для числових множин”

## **Мета роботи:**

Ознайомитись на практиці із основними поняттями теорії множин, навчитись будувати діаграми Ейлера-Венна операцій над множинами, використовувати закони алгебри множин, освоїти принцип включення-виключень для двох і трьох множин та комп’ютерне подання множин.

## **Теоретичні відомості:**

### **2.1. Основні поняття теорії множин. Операції над множинами**

**Множина** – це сукупність об’єктів, які називають елементами.

Кажуть, що множина  $A$  є **підмножиною** множини  $S$  (цей факт позначають  $A \subseteq S$ , де  $\subseteq$  – знак нестрогого включення), якщо кожен її елемент автоматично є елементом множини  $S$ . Досить часто при цьому кажуть, що множина  $A$  міститься в множині  $S$ .

Якщо  $A \subseteq S$  і  $S \neq A$ , то  $A$  називають **власною (строгою, істинною) підмножиною**  $S$  (позначають  $A \subset S$ , де  $\subset$  – знак строгого включення). Дві множини  $A$  та  $S$  називаються **рівними**, якщо вони складаються з однакових елементів. У цьому випадку пишуть  $A=S$ .

Якщо розглядувані множини є підмножинами деякої множини, то її називають **універсумом** або **універсальною множиною** і позначають літерою  $U$  (зауважимо, що універсальна множина існує не у всіх випадках). Множини як об’єкти можуть бути елементами інших множин, Множину, елементами якої є множини, інколи називають **Сімейством**.

Множину, елементами якої є всі підмножини множини  $A$  і тільки вони (включно з порожньою множиною та самою множиною  $A$ ), називають **булеаном** або **множиною-степенем** множини  $A$  і позначають  $P(A)$ .

**Потужністю** скінченної множини  $A$  називають число її елементів, позначають  $|A|$ .

Множина, яка не має жодного елемента, називається **порожньою** і позначається  $\emptyset$ .

Вважається, що порожня множина є підмножиною будь-якої множини, а також  $A \subseteq A$ .

Множина всіх підмножин множини  $A$  називається **булеаном** і позначається  $P(A)$ . Потужність скінченної множини дорівнює кількості її елементів, позначається  $|A|$ . Потужність порожньої множини дорівнює 0.

## Варіант № 5

### Завдання 1:

1. Для даних скінчених множин  $A = \{1,2,3,4,5,6,7\}$ ,  
 $B = \{4,5,6,7,8,9,10\}$ ,  $C = \{1,3,5,7,9\}$  та універсума  $U = \{1,2,3,4,5,6,7,8,9,10\}$  знайти множину, яку задано за допомогою операцій: а)  $A \cap B \cup C$ ; б)  $\neg A \Delta \neg C$ .

для  $U = \{1,2,3,4,5,6,7,8,9,10\}$

а)  $A \cap B \cup C = (1011111010)$

б)  $\neg A \Delta \neg C = (0101010010)$

2. На множинах задачі 1 побудувати булеан множини.

$C \setminus (\neg A \cup \neg B) \cap C$ . Знайти його потужність.

$$P(C \setminus (\neg A \cup \neg B) \cap C) = \{\emptyset, \{5\}, \{7\}, \{5,7\}\}$$

$$|P(C \setminus (\neg A \cup \neg B) \cap C)| = 2$$

3. Нехай маємо множини:  $N$  –множина натуральних чисел,  $Z$  – множина цілих чисел,  $Q$  – множина раціональних чисел,  $R$  – множина дійсних чисел;  $A, B, C$  – будь-які множини. Перевірити які твердження є вірними (в останній задачі у випадку невірного твердження достатньо навести контрприклад, якщо твердження вірне – навести доведення):

- а)  $3 \in \{\{1, 2, 3\}, 4\}$ ; б)  $Z \subset N$ ;
- в)  $Q \cap Z \subset R \setminus N$ ; г)  $Q \setminus Z \subset R \setminus N$ ;
- д) якщо  $A \subset B \wedge A \subset C$ , то  $A \subset B \cap C$ .

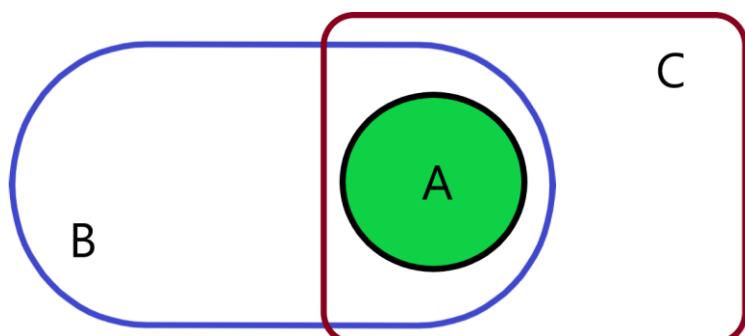
а)  $3 \in \{\{1, 2, 3\}, 4\}$  – хибне ( $3$  не належить множині  $\{\{1, 2, 3\}, 4\}$ )

б)  $Z \subset N$  – хибне (множина цілих чисел не є підмножиною множини натуральних чисел)

в)  $Q \cap Z \subset R \setminus N$  – хибне (множина цілих чисел не є підмножиною множини дійсних ненатуральних чисел)

г)  $Q \setminus Z \subset R \setminus N$  – вірне (множина раціональних нецілих чисел є підмножиною множини дійсних ненатуральних чисел)

д)  $A \subset B \wedge A \subset C$ , то  $A \subset B \cap C$  – вірне, тому що:



4. Логічним методом довести тотожність:

$$A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C).$$

Для доведення скористаємось законами алгебри множин.

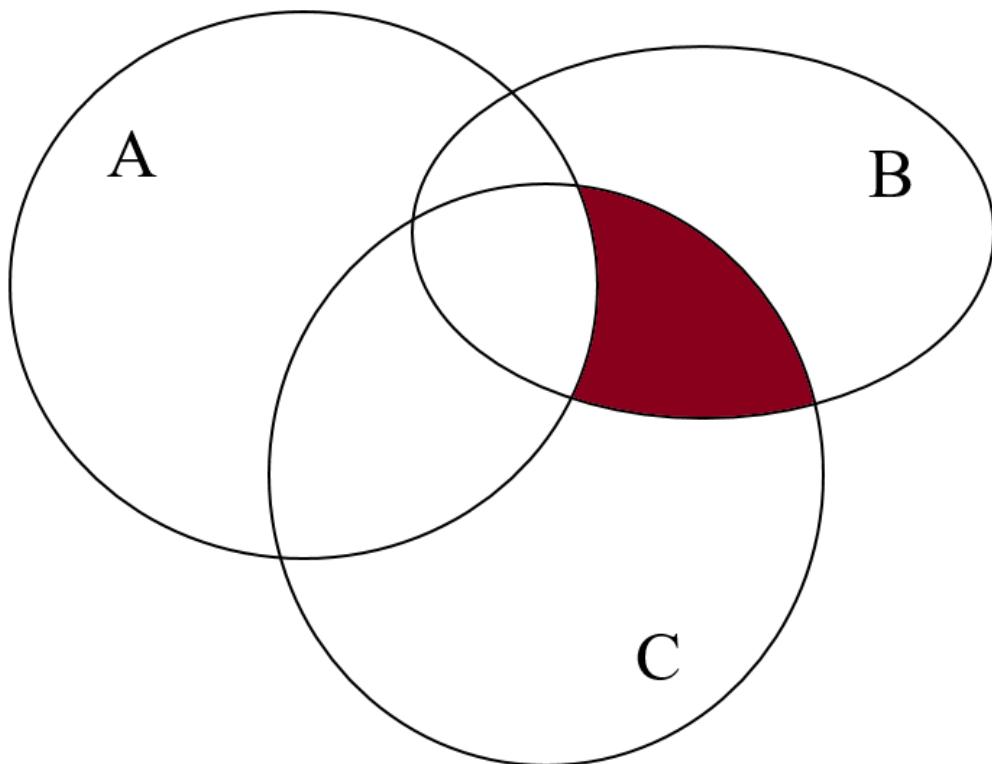
$$A \setminus (B \cup C) = A \cap \neg(B \cup C) = A \cap (\neg B \cap \neg C) = A \cap \neg B \cap \neg C$$

$$(A \setminus B) \cap (A \setminus C) = (A \cap \neg B) \cap (A \cap \neg C) = A \cap \neg B \cap A \cap \neg C = A \cap \neg B \cap \neg C$$

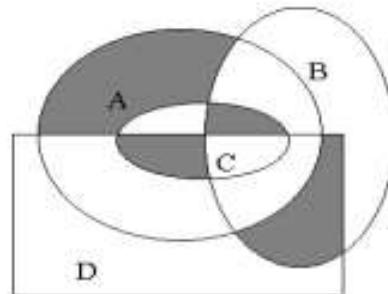
Отже,  $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$ .

5. Зобразити на діаграмі Ейлера-Венна множину  $B \cap (A \Delta (C \setminus B)) \setminus A$ .

$B \cap (A \Delta (C \setminus B)) \setminus A = (B \cap C) \setminus A$ , тому діаграма матиме наступний вигляд:



6. Множину зображену на діаграмі. Записати її за допомогою операцій.



$$(\neg A \cap B \cap C \cap \neg D) \cup (A \cap \neg B \cap \neg C \cap \neg D) \cup (\neg B \cap C \cap D \cap \neg D) \cup (B \cap C \cap \neg D \cap \neg D)$$

## Завдання 2:

### Програма:

Ввести з клавіатури дві множини символьних даних. Реалізувати операцію об'єднання над цими множинами. Вивести на екран новоутворену множину. Знайти програмно булеван цієї множини.

```
1 #include <math.h>
2 #include <cs50.h>
3 #include <ctype.h>
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7
8 //DECLARING PROTOTYPES///////////////////////////////
9
10 char* concat(char *s1, char *s2); //, char *s3
11 void rec(string mas, int num, string s);
12
13 //MAIN/////////////////////////////
14
15 int main(void){
16
17     int ar1_len = 0;
18     int ar2_len = 0;
19     int mrs_len = 0;
20     char rs1[50];
21     char rs2[50];
22     char mrs[50];
23
24 //GETTING THE SET ELEMENTS///////////////////
25
26     printf("Enter the elements of set #1\n");
27     string ar1 = GetString();
28     printf("Enter the elements of set #2\n");
29     string ar2 = GetString();
30
31 //CLEANSING FROM NON-ALPHABETICAL CHARS///////////////////
32
33     for (int i = 0, l = strlen(ar1); i<l; i++){
34         if (isalpha(ar1[i])){
35             rs1[ar1_len] = ar1[i];
36             ar1_len++;
37         }
38     }
39
40     for (int i = 0, l = strlen(ar2); i<l; i++){
41         if (isalpha(ar2[i])){
42             rs2[ar2_len] = ar2[i];
43             ar2_len++;
44         }
45     }
46
47 //PROCESSING AND CLEANSING FROM DUPLICATE CHARS///////////////////
48
49     for (int i_1 = 0; i_1 < ar1_len; i_1++){
50         bool unique = 1;
51
52             for(int i_u = 0; i_u < mrs_len; i_u++){
53                 if (rs1[i_1] == mrs[i_u]){
54                     unique = 0;
55                 }
56             }
57
58             if (unique){
59                 mrs[mrs_len] = rs1[i_1];
60                 mrs_len++;
61             }
62     }
63
64     for (int i_2 = 0; i_2 < ar2_len; i_2++){
65         bool unique = 1;
66
67             for(int i_u = 0; i_u < mrs_len; i_u++){
68                 if (rs2[i_2] == mrs[i_u]){
69                     unique = 0;
70                 }
71             }
```

```

73         if (unique){
74             mrs[mrs_len] = rs2[i_2];
75             mrs_len++;
76         }
77 }
78
79 //OUTPUT OF SET///////////////////////////////
80
81 printf(" Result of union| is { ");
82 if (mrs_len != 0){
83     for (int i = 0; i<mrs_len; i++){
84         printf("%c", mrs[i]);
85         if (i != (mrs_len - 1)){
86             printf(", ");
87         }
88     }
89 }
90 printf(" }\n");
91
92 //OUTPUT OF BOOLEAN///////////////////////////
93
94 printf(" Boolean is { ");
95 if (mrs_len != 0){
96     rec(mrs, 0, "");
97 }
98 printf(" }\n");
99
100 }
101
102 //STRING CONCATENATION FUNCTION///////////////////
103
104 char* concat(char *s1, char *s2){ //, char *s3
105     size_t len1 = strlen(s1);
106     size_t len2 = strlen(s2);
107 //     size_t len3 = strlen(s3);
108
109     char *result = malloc(len1 + len2 + 1); // + len3
110
111     if (!result){
112         return NULL;
113     }
114
115     memcpy(result, s1, len1);
116     memcpy(result + len1, s2, len2 + 1);
117 //     memcpy(result + len1 + len2, s3, len3+1);
118
119     return result;
120 }
121
122
123 //BOOLEAN FINDER FUNCTION///////////////////
124
125 void rec(string mas, int num, string s){
126     if(num == strlen(mas)){
127         printf("{%s} ", s);
128         return;
129     }
130
131     rec(mas, num+1, s);
132     char mn[2] = "\0";
133     mn[0] = mas[num];
134     s = concat (s, mn); //, " "
135     rec(mas, num+1, s);
136 }

```

## Висновки:

Я ознайомився на практиці із основними поняттями теорії множин, навчився будувати діаграми Ейлера-Венна для операцій над множинами, використовувати закони алгебри множин, освоїв принцип включення-виключень для двох і трьох множин та комп'ютерне подання множин.